Australasian Transport Research Forum 2016 Proceedings
16-18 November 2016, Melbourne, Australia
Publication Website: http://www.atrf.info

Machine Learning Approach for Self-Learning Eco-Speed Control

Hasitha Dilshani Gamage[1] , Dr. Jinwoo (Brian) Lee[2]
Civil Engineering and Build Environment School, Queensland University of Technology, Brisbane, Australia
[1] h.dabaregamage@qut.edu.au, [1] brian.lee@qut.edu.au

Email for correspondence: brian.lee@qut.edu.au

Abstract

Significant fuel consumption occurs by traffic signals due to their periodic disruption of traffic flow. This paper proposes a Q-learning based vehicle speed control algorithm to minimise the fuel consumption in the vicinity of an isolated signal intersection. Q-learning is a self-learning algorithm that learns the optimal control action(s) based on the trial-and-error approach. The speed control algorithm is trained in the Aimsun microsimulation platform under varying traffic signal and arrival speed conditions. The training and validation of the algorithm are conducted under the single vehicle scenario where only one control vehicle presents in the intersection approach. A comprehensive parametric analysis provides fine-tuning of the Q-learning parameters and the impact of parameter settings on the algorithm's performance and convergence. Using the chosen parameter setting, the performance of the algorithm is demonstrated in comparison with a vehicle velocity profile for a baseline scenario where the speed control is disabled. The simulation results indicate that the algorithm can reduce the vehicle's fuel consumption by 15.78% by adopting the suggested driving speeds.

1. Introduction

In recent years, significant efforts have been put into identifying effective countermeasures to improve vehicle fuel efficiency while reducing negative environmental impacts. Driving behaviour has a significant impact on a vehicle fuel consumption and emission exhaust (Yang et al., 2012). By following the fuel optimal speed trajectory can reduce vehicle fuel consumption significantly. Eco-driving is one of the most promising practical solutions that can substantially reduce adverse environmental impacts of traffic. This is being achieved by enhancing or optimising driving manoeuvre while maintaining vehicle's dynamic performances. Maintaining constant velocity while minimising unnecessary acceleration and deceleration is the fundamental principle of eco-driving (Barth et al., 2011).

Recent improvements in Intelligent Transport Systems (ITS) have led to the development of smart and dynamic speed guidance systems. In Europe, some public transit vehicles communicate with the traffic lights, which gives a positive start towards developing more advanced applications (Koenders & Vreeswijk, 2008). Researchers have identified that real-time speed guidance using traffic signal information can improve long-term benefits of eco-driving up to 20% (Barth & Boriboonsomsin, 2009). Linear Programming, Dynamic Programming and Heuristic optimisation such as Genetic Algorithm have been adopted to derive the fuel optimal velocity profiles (Chen et al., 2014; Kamal et al., 2010a; Kamalanathsharma & Rakha, 2013; Wu et al., 2011). Besides these traditional methods, there is a growing trend in applying Artificial Intelligent (AI) methods such as Reinforcement Learning (RL) to transport engineering applications due to their wide applicability and capability of performing well in complex situations.

The core concept behind the Reinforcement Learning (RL) is based on human learning and decision-making process. The learner or the decision maker is called as the "Agent". Where the agent self-learn the optimal control action (optimal policy) by direct trial-and-error interaction with the environment using the positive or negative rewards that are given to access the desirability of performed action under a particular environmental condition. Once the agent gain required amount of learning, agent can apply optimal control actions directly using the learnt policy. The main aim of this study is to conduct a comprehensive parametric analysis to fine- tune the Q-learning parameters and find the impact of parameter setting on algorithm performances, and convergence to the optimal control policy as there is no comprehensive study that has been conducted yet. The agent's performances with chosen parameter setting has been compared with vehicle velocity profile for baseline case (i.e. Vehicles that do not have Q-learning based speed control) in AIMSUN simulation environment to demonstrate the effectiveness of the proposed Q-learning algorithm for the application of optimal speed control of vehicles.

2. Literature Review

Considerable fuel wastage occurs at signalised intersections due to the sudden response for changing traffic signals. Vehicles can significantly reduce unnecessary acceleration, deceleration, and idling behaviour if traffic signal information is present when making a decision. Significant advancements in ITS sector has made it possible to transfer information from infrastructure to vehicles or even between vehicles. As a result, recent studies have proposed Driver Assistance Systems (DAS) which utilises available real-time traffic signal

data. DAS has a positive impact on reducing fuel consumption by advising the driver with relevant real-time information, which results in smooth driving and on time arrival at intersections.

Early researchers on eco-driving is oriented towards providing simple speed guidance to drivers without providing optimal driving speeds considering prevailing traffic and traffic control conditions. For an example, Wu et al. (2011) proposes an Advance Driving Alert System (ADAS) which displays a simple message if the vehicle's estimated travel time falls within the range of red signal timing. Proposed system is compared with Changeable Message Signs (CMS), and up to 40% fuel saving is achieved under hypothetical conditions. Recent studies have proposed dynamic eco-driving strategies by providing optimal speed control suggestions to minimise excessive fuel consumption. Mandava et al. (2009) develops an algorithm using constrained optimisation technique to determine minimum acceleration level to achieve the target velocity, and Barth et al. (2011) develops a dynamic velocity planning algorithm which constrict vehicle acceleration/deceleration to reduce fuel consumption and emissions. However, both of these research attempts to oversimplify the fuel consumption function by only using acceleration rate, or timely arrival at an intersection rather than using an explicit objective function to minimise the fuel consumption.

Another related studies have employed stochastic optimisation techniques such as Dynamic Programming (DP) (Hellström et al., 2010; Kamalanathsharma & Rakha, 2013; Mensing et al., 2011) and open-loop deterministic optimisation control techniques such as Model Predictive Control (MPC) (Asadi & Vahidi, 2011; Kamal et al., 2010b), which requires high computational cost and thus inappropriate to real-time speed control (Zhang et al., 2015). Other than above techniques, heuristic optimisation, including Genetic Algorithm (GA) has accommodated to derive the optimal eco-driving trajectory at signalised intersection (Chen et al., 2014). However, DP assumes an accurate model of the traffic environment which is challenging due to the stochastic and dynamic nature of traffic environment. Hence, status of traffic condition must frequently be monitored, and the driving speed of vehicles must be adjusted appropriately to changing traffic status (Sutton & Barto, 1998).

Recent literature shows that Artificial Intelligent techniques such as Reinforcement Learning (RL) has been adopted for transport applications (Abdulhai & Kattan, 2003). However, incorporating RL into transport section is quite new, and more research possibilities needs to be explored and examined due to the advantages of this technique such as, unsupervised nature, ability of self-learning, and low computational cost (Abdulhai & Kattan, 2003). Possible applications of RL (Q-learning technique), to provide the fuel optimal speed suggestions at signalised intersections is yet to be explored. Therefore, finding a complete study to identify the most effective Q-learning design parameter configuration is quite challenging which is addressed in this paper.

In the literature, several microscopic fuel consumption models have been used to estimate vehicle fuel consumption (Barth et al., 2011; Chen et al., 2014; De Nunzio et al., 2013; Kamal et al., 2010b; Kamalanathsharma, 2012). These models calculate individual fuel consumption based on second-by-second vehicle trajectory data. Vehicle velocity and acceleration profiles are the main deciding factors on fuel consumption. It is prohibitively difficult to test the proposed eco-speed control algorithm in field experiments. Therefore, the best possible way to test the performances of proposed algorithm is by conducting simulation studies. In this study, we use AIMSUN (Advanced Interactive Microscopic Simulator for Urban and Non-

Urban Networks) microscopic traffic simulation software to conduct experiments. The fuel consumption model embedded in AIMSUN is used to calculate fuel consumption of the vehicle.

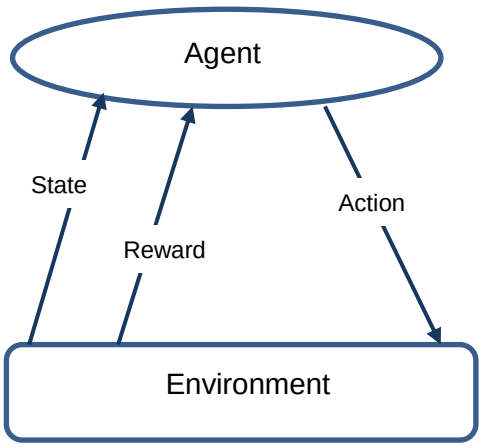
3. Methodology

3.1 Reinforcement Learning

Reinforcement Learning (RL) is a close-loop autonomous process that is inspired by the human learning behaviour and decision-making process. The agent does not need any external supervision to learn how to solve the optimal control problem as RL is a trial-and-error based approach. In RL the autonomous agent learns the best control action by sensing its` environment by selecting an action, and receiving a reward or penalty; a scalar value which describes the success or the failure of the performed action. The control algorithm also have the ability to improve the performances using system feedbacks. RL assumes that the system dynamics can be formulated based on Markov Decision Process (MDP) mathematical framework. MDP is a discrete time stochastic optimal control process. The MDP is a tuple <S,A,p,R> where the elements are; state, actions, transition probabilities, and transition reward respectively. The key feature of Markov process is called as Markov property where the presence is the independence of the past. Therefore controlled agent should have the ability to describe the current situation without knowing all the past information.

Q-learning is one of the most used techniques among other RL techniques due to its model-free nature. Q-learning was introduced by Watkins (1989) where this technique allows learning to accomplish an arbitrary task from experience gained by direct interaction with the environment. Everything that agent interact with, is called a state, where states represents some situations of the environment in correspondence to the control problem. RL consists of series of episodes (n) where each episode consists with a sequence of state-action pairs $(s_t^n, a_t^n), t \in \{0,1, \dots \dots T\}$, with T steps. Each episode starts with an initial state and ends when the control agent reaches to its` terminal state or satisfies the termination criteria. During the learning period, at each episode, agent perceives the current state s_t of the environment, choose an action a_t. The action results in changes in the state of the environment, which results in agent to move into a new state. As mentioned above, the desirability of executed action while being in a given state space is assessed by the scalar reward. Figure 1 shows the agent-environment interaction.

Figure 1. Agent-environment interaction



Agent's mapping from state to action is called as policy π. The policy is improved iteratively as a result of agent's experience. Many possible trials are executed during the training phase to confirm that the agent has learnt from enough experiences. The control problems` ultimate goal is to find the optimal policy π^* which determines the best control action when the agent is in a particular state.

$$\pi*(s) \in argmaxa \in AQt(s,a) \quad (1)$$

The value associates with that state-action pair is updated with its current value $Q(s_t, a_t)$, instance reward that receives for the executed action r and with the expected return starting from that state$Q\ (s_{t+1}, a_{t+1})$. The data structure of Q-value is a matrix, where each cell element represents the corresponding value of taking a particular action, for being in a particular state. The state-action value is maximised when agent is following the optimal policy. The Q-value table is updated by using the following one-step equation,

$$Q_t^N(s', a) = Q_t^{N-1}(s', a) + \alpha_t\{[r_t(s', a) + \gamma max_{a_{t+1}} Q_{t+1}^{N-1}(s', a)] - Q_t^{N-1}(s', a)\} \quad (2)$$

Here γ is the discount factor$(0 \leq \gamma \leq 1)$, and α is the learning rate$(0 \leq \alpha \leq 1)$.

Algorithm 1. Pseudo code of Q-learning is as follows,

Input: set of states$S$, set of actions$A$, reward R
Input: Discount rate (γ), Learning rate (α) and action selection policy parameter(ε)
Initialise $Q(s, a)$ arbitrarily for every state s and every action a
For each episode **do**

Initialise s

Repeat {for each step of the episode}

Choose a from s based on the policy derived
(e.g. $\varepsilon\ - greedy$)
Take action a, observe r, s.'
$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma max_\alpha Q(s', a') - Q(s, a)]$
$\pi(s) \leftarrow \arg max_{a \in A} Q(s, a)$
$s \leftarrow s'$;
Until s is terminal

End for

3.2 Algorithm Design

3.2.1 State and Action Space

State space represents characteristics of the environment which are useful to solve the problem. In our case, vehicle's current speed, positions and traffic signal information are important factors when deciding its fuel optimal speed trajectory. The vehicle speed controls are applied considering two pre-defined decision points namely at 300m & 150m from the intersection. Capturing all information of a car entering to the 300m from the stop line, the state space of a vehicle speed control problem can be written as,

$$S_i = \{v_i, d_i, t_i\} : v_i \in V,\ d_i \in D \qquad \forall i \in \{1,2,3 \dots \dots . . n\}$$

Where v is the current speed of i th vehicle (m/s) and d is the current distance to the stop line in (meters). The t = reaming time in the current phase. At the beginning, all possible ranges of velocities (v) are discretised as follows,

$$v_i = 2.n\ ,\ \ n = 1,2,3,4 \dots . . n$$
$$v_{min} \leq\ v_i \leq\ v_{max}$$

Here v_{min} is set to 20km/h (5.5 m/s) and v_{max} is set to 60km/h (16.6 m/s). This study assumes that the vehicle accelerates or decelerates or maintains its current speed as guided by algorithm and cruise with that velocity until it reaches the next decision point. The maximum acceleration and deceleration range varies from +2.7ms^{-2} to -2.7ms^{-2}. The remaining time in current phase is discretised using 1 second intervals. Total cycle length is 60 seconds within the range of 30seconds of green and 30 seconds of red. Using the information related to detected state, the control agent chooses an action. The action space consists with target speeds which can be achieved by either acceleration, deceleration or cruising while obeying to constraints imposed by road way speed limit and maximum and minimum acceleration/deceleration limits. The target speeds (possible actions) are also discretised with the same approach used for current speed discretisation while keeping the maximum and minimum roadway speed limits to 20km/h and 60km/h respectively.

3.2.2 Reward Function

Reward function is one of the most important parameters which determines the success of learning of the agent. Because the only signal used by the agent to learn from performed actions is the reward. The main goal of speed control agent is to minimise the fuel consumption along the trajectory. Therefore, in this study the reward function is inversely proportional to the cumulative fuel consumption, experienced by the vehicle between two successive decision points. In order to confirm that the agent discharges within the earliest possible green time, a positive scalar value is added to the reward function and also a negative scalar value is given if the agent discharge the intersection during the red time. If the reward has a higher value, this means that the fuel consumption is reduced as a result of chosen action. If the reward value is low, this means the executed action has consumed a higher amount of fuel.

3.3 Action Selection Strategy

In RL, the agent's goal is to maximise the total amount of reward it receives over time. The primary challenge that arises during the action selection process is, when the agent tries to choose the action with the highest reward value in order to maximise the short-term reward. However, the agent needs to explore new actions for better performances in the future. There are few action selection policies to balance the exploration and exploitation in the RL agent. Ɛ- Greedy is one of the well-known action selection policy.

Ɛ- Greedy

The simplest action selection rule is to choose the actions with highest estimated state-action value which is called the greedy action. Therefore, in exploitation agent tends to choose the actions based on what agent knows best in the environment. Making a balance between exploitation and exploration by behaving most of the time greedily while choosing a uniform random action with small probability ε, can be identified as the best alternative way to discover

better action selections in the future. The random action selection is independent of the action-value estimation. This method is called as ε- Greedy which is a popular yet simple alternative technique. The advantage of this approach is that when the number of training episodes are increased, agent visits every state-actions pair which guarantees convergence of the Q_t (a) to optimal policy; Q^*(a). The ε- greedy method eventually performs better as it has a higher chance of identifying the optimal action as a result of continuing exploration.

$$\pi(s_t) = \begin{cases} random\ action\ a \in A(s_t) & if\ \rho < \epsilon \\ argmax_a Q(s_t, a) & otherwise \end{cases}$$

Here, $0 \leq \rho \leq 1$ is a uniform random number generated at each decision point. There are two main types of ε- Greedy methods. One is by employing a constant value for ε (e.g.0.1) and the other is by employing the gradually decaying exploration rate. The latter technique helps the agent learn better polices where, at the beginning the agent explores by trying a range of random actions to familiarise with the environment and, towards the end of learning period the agent tends to exploit more as a result of converges to the optimal policy. Jacob and Abdulhai (2005) suggested this gradually decreasing exploration rate by an exponentially decreasing function as below,

$$\varepsilon = e^{-En} \qquad (3)$$

Where:
n = the number of iterations or the age of the agent
E = exploitation parameter
E is a constant between 0 and 1 that determines how much probability depends on the age and relative Q values. This selects the "best" action (greedy action) with probability P = 1-e^{-En}.

3.4 Experimental Set-Up

There are two main phases in the experimental setup: the training phase (learning phase) and the implementation phase. The training is conducted in AIMSUN microscopic environment where the test-bed consists of an isolated signalised intersection. A typical car is chosen as the training agent. The Q-learning agent algorithm is developed using python programing language and the simulation environment is developed using AIMSUN API. As the inputs set of states S, set of actions$A$, reward$R$, discount rate (γ), learning rate(α) , and action selection policy parameter(ε) are defined as mentioned in section 3.2 and 3.3. During the initial training phase, the agent has no knowledge about which actions needs to be executed at various state conditions. Therefore, the Q-table entries are initialised with zeros. Since the number of state-action pairs is relatively small in this application, elements of the Q-value function are stored in a lookup table (Q-table). After executing each action, the Q-value for each state-action pair is computed by using Q-learning equation 2 in section 3.1, and the relevant position is updated. Here, the agent's goal is to choose the actions that minimise the vehicle fuel consumption under varying state-space conditions (i.e. varying traffic signal and arrival speeds).

Theoretically each state-action pair needs to be visited infinite number of time. In reality, visiting each state-action pair infinitely is not possible. However, the QL algorithm can still

converge to a reasonable approximation of the optimal policy after some pre-defined criteria. This study assumed the convergence of the algorithm when the absolute value of Q-value function in iteration N and iteration N-1(TD_{error}) is oscillating with a small variation for few consecutive iterations.

$$\Delta Q_t^N = \left| Q_t^N(s', a) - Q_t^{N-1}(s', a) \right| \qquad \forall\, t, s', a \qquad (4)$$

After Q-learning agent builds up the knowledge with a sufficient number of iterations, the implementation of the proposed algorithm is undertaken at an isolated signalised intersection in AIMSUN microsimulation environment. The intersection is modelled to operate considering pre-timed traffic signal timing with 60 seconds of cycle length.The proposed Q-learning based vehicle speed control algorithm, is compared with the baseline case. Vehicles` velocity profiles for baseline case follows the average driving behaviour while suddenly responding when the driver visually recognises the traffic signal change (i.e. No exchange of traffic signal information with oncoming traffic is assumed). The maximum speed limit of the road segment for both scenarios are 60km/h (16.67 m/s). The minimum cruising speed of the controlled vehicles is set to 20km/h (5.55m/s). For the comparison purposes, this study determined the fuel consumption for both baseline and controlled scenarios using fuel consumption model plug-in to AIMSUN.

4. Simulation Results and Discussion

4.1 Learning Performances

The number of state-action pair visits during the learning phase has a significant impact on the performances of the learning agent. The learning efficiency and the effectiveness of the agent can be demonstrated through a Q-table as it stores data for all state-action pairs. Figure 2 shows the Q-value table after a small number of iterations.

Figure 2. Q-value Surface Plot after Small Number of Iterations

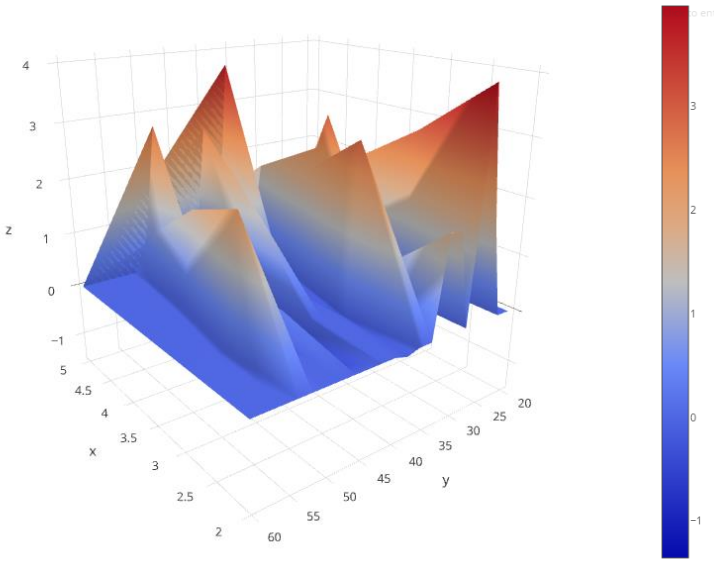
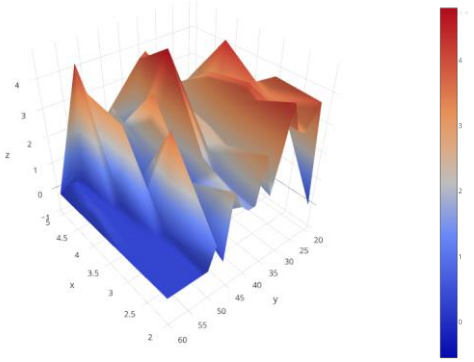
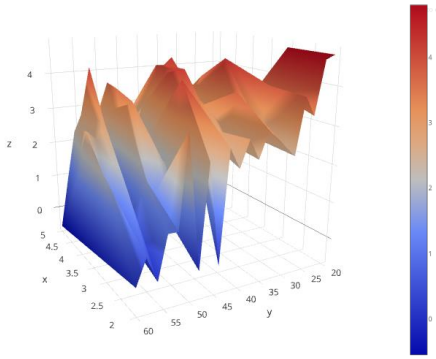


Figure 2 represents Q-value surface variation for few set of sample state-action pairs after a small number of iterations (1000 iterations) during the learning process. Z-axis represents the Q-value, X- axis represent the sample state and Y-axis represent the actions. With a careful examination of the values, it is visible that the most values stay within the 0 (blue colour) which displays very limited learning process or the agent has not explored most of the states during the learning period. Therefore value updating has not occurred. When compared with the size of chosen state-space, only around 30 % has visited during learning. Indicating more state-action pair visits is needed during the learning process to find the true optimal policy. In the initial phase, agent needs to explore more and over the time (once enough number has visited during each training episode) agent tends to choose more greedy actions. Therefore, having sufficient number of training is crucial during the learning period.

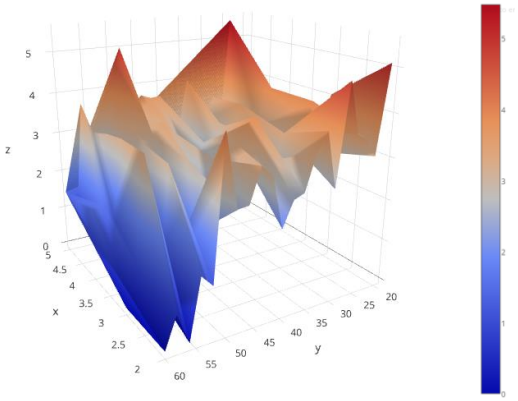
Figure 3: Q-surface Plot after High Number of Iterations



(a) 5000 iterations



(b) 20 000 iterations



(b) 30 000 iterations

The illustration of figure 3 represents the Q-value variation of the agent over more number of iterations. Z-axis represent the Q-value, X- axis represent the sample state and Y-axis represent the actions. In conclusion, for the selected sample size , 66%, 89% and 94% of state-action pairs have been visited during the 5000, 20,000 and 30,000 iterations respectively. Higher number of iterations made the agent explore more state-action pairs (Q-values), indicated by dark orange zone. Unvisited state-action pairs have reduced significantly with increasing number of iterations. Also the comparatively higher variation of the results during

the learning process is an indication of agent's exploration of each state-action pair before converging to the optimal policy.

4.2 Action Selection Parameter

As mentioned in section 3.3 action section policy has a significant impact on the agent's learning process and employing most appropriate action selection policy, which guarantees agent's best performance. In consequence this study conducted an extensive study to find the impact of ϵ −greedy action selection policy on convergence of the agent. The initial analysis is conducted to establish the most suitable exploration rate for the ε in ε- greedy policy. The discount factor is set at 0.8 and the learning rate at 0.7. The constant setting is tested with the value of ε at 0.1, 0.2, and 0.3. The results are compared while considering two performance measures of convergence, including convergence speed (number of simulations required to converge), and the precentage of each state-action pair visited.

Figure 4: Constant Exploration Rates

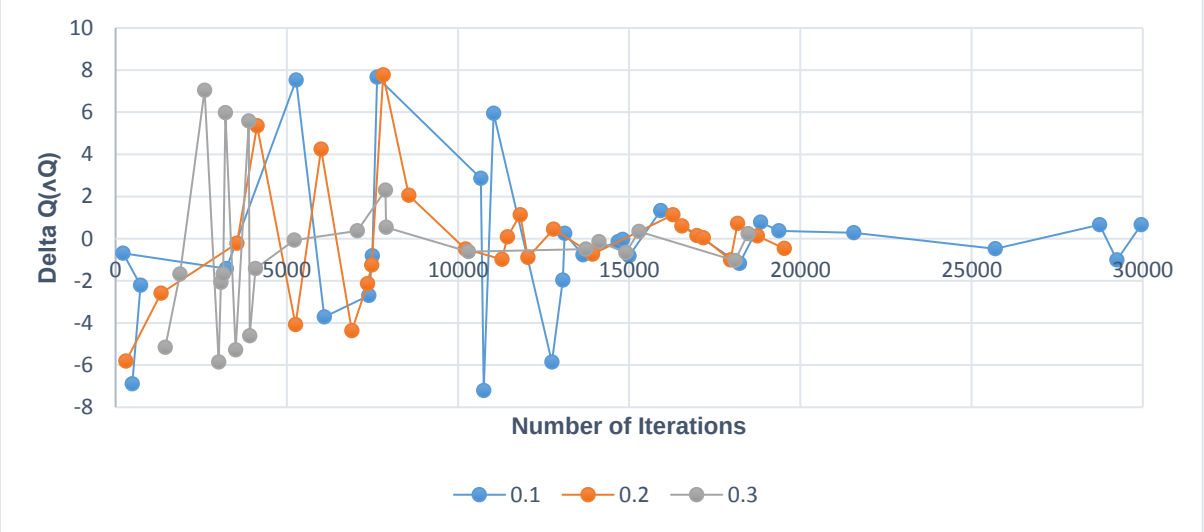


Table 1: Results With Constant Exploration Rates

Sigma Value	Convergence Speed	% Total State Action Pair Visited
0.1	18901	78.2
0.2	12402	82.5
0.3	10304	93.7

Figure 1 represents the agent's convergence with constant ε for a sample state-action pair. The delta Q-value represents the variation between Q-value at iteration N and (N-1). When using the ε = 0.1 agents chooses the greedy action most of the time rather than choosing a random action with 0.1 probability. However, it is required to visit all state-action pairs (theoretically infinite number of times) at least few number of times to guarantee to converge to the optimal policy. With the careful examination, it is noticed that the agent did not visit

some state-action pairs at all with ε = 0.1 value. As shown in the Table 1, the constant exploration rate ε = 0.3 captures more state-action pairs during the learning period (93.7%). It is assumed that the algorithm converged when the absolute difference between two successive Q-values oscillate with the small variation for a few consecutive iterations. For the ε = 0.3 the algorithm converges with around 10304 iteration. Next, the Q-learning agent is trained using a gradually decaying exploration rate.

Figure 5: Results With Decaying Exploration Rates

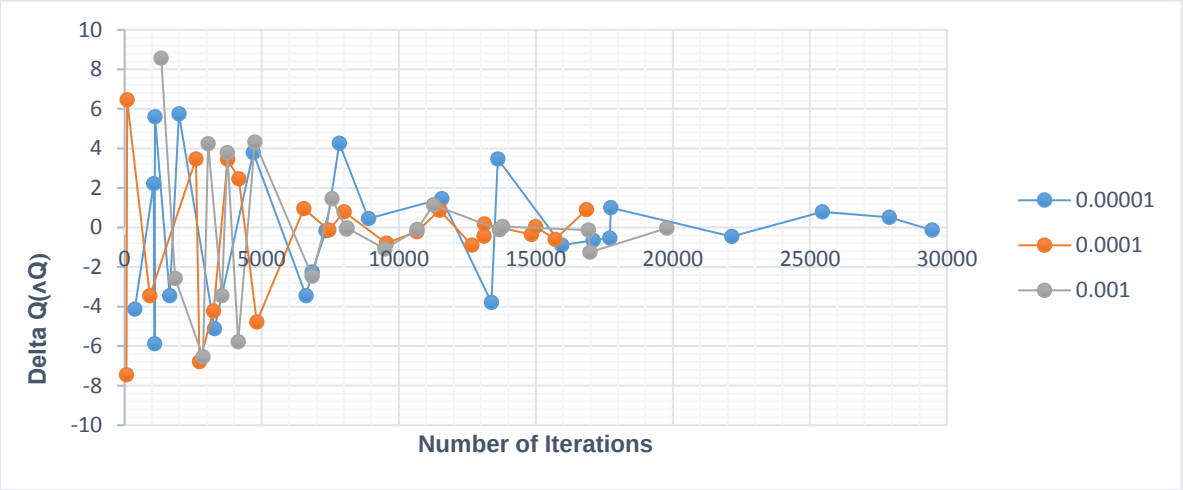


Table 2: Convergence with Decaying Exploration Rates

Sigma Value	Convergence Speed	% Total State Action Pair Visited
E = 0.00001	15945	73.4
E = 0.0001	9551	95.2
E=0.001	13673	86.9

To test the decaying setting, the value of E as in equation 3 in section 3.3 is set to have a varying rate at 0.001, 0.0001, and 0.00001. With decaying *ε,* the exploration rate decreases as the number of iteration increases. At the initial stages, the agent tends to explore more frequently to gain more knowledge, and eventually, the agent chooses more greedy actions as the iteration number becomes larger. According to the sensitivity analysis test results, agent visits more state action pairs by the time of convergence to the optimal policy when using gradually decreasing the rate of exploration (E= 0.0001). Here, the convergence of the agent is demonstrated by decreasing the variation of Q-values (delta Q) as the age of the agent increases over the iteration. Therefore, gradually decreasing exploration with E=0.0001 is selected for this study as it provides the best performances over other values.

4.3 Performance Demonstration

The fine-tuned Q-learning agent was trained in the microsimulation testbed. The trained agent was then tested to demonstrate its performance by comparing the fuel consumption of test

vehicles with and without the speed control. We tested total 200 vehicles in two different instances: the first instance with the speed control and the second instance without the speed control under the exactly identical traffic signal status and the vehicle arrival speed.

The test results indicate that the speed control algorithm is able to reduce the fuel consumption by 15.78% as presented in Table 3.

Table 3: Fuel Consumption Comparison

	Controlled		Uncontrolled		Improvement
Fuel (ml)	Average	S.D.	Average	S.D.	15.78%
	8.33	1.88	9.89	2.89	

5. Conclusion and Future Recommendation

This paper proposes a Q-learning based speed control algorithm to minimise the vehicles' fuel consumption in signalised intersections. A comprehensive analysis was conducted to fine-tune the Q-learning parameters and the impact of parameter settings on the algorithm's performance and convergence. The analysis results indicate that visiting all the state-action pairs during the learning phase is important to find the true optimal policy. This study discovered gradually decreasing rate of exploration (E=0.0001) performs best over other exploration rates.

The performance of the agent with the chosen parameter setting was evaluated by comparing the vehicles' fuel consumption with and without the speed control. The test showed promising results by reducing the averaged vehicle fuel consumption by 15.79%.

The current study will be further extended to reflect more realistic driving environments (i.e. presence of other traffic). Moreover, the various Q-learning input parameter settings will be applied to further improve the algorithm performances.

References

Abdulhai, B., & Kattan, L. (2003). Reinforcement learning: Introduction to theory and potential for transport applications. *Canadian Journal of Civil Engineering, 30*(6), 981-991.

Asadi, B., & Vahidi, A. (2011). Predictive Cruise Control Utilizing Upcoming Traffic Signal Information for Improving Fuel Economy and Reducing Trip Time. *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, vol.19*.

Barth, M., & Boriboonsomsin, K. (2009). Energy and emissions impacts of a freeway-based dynamic eco-driving system. *Transportation Research Part D: Transport and Environment, 14*(6), 400-410.

Barth, M., Boriboonsomsin, K., Mandava, S., & Haitao, X. (2011). *Dynamic ECO-Driving for Arterial Corridors*. Paper presented at 2011 IEEE Forum on Integrated and Sustainable Transportation Systems, Vienna, Austria.

Chen, Z., Zhang, Y., Lv, J., & Zou, Y. (2014). Model for Optimization of Ecodriving at Signalized Intersections. *Transportation Research Record: Journal of the Transportation Research Board*(2427), 54-62.

De Nunzio, G., Canudas de Wit, C., Moulin, P., & Di Domenico, D. (2013). Eco-driving in urban traffic networks using traffic signal information. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on* (pp. 892-898): IEEE.

Hellström, E., Åslund, J., & Nielsen, L. (2010). Design of an efficient algorithm for fuel-optimal look-ahead control. *Control Engineering Practice, 18*(11), 1318-1327.

Jacob, C., & Abdulhai, B. (2005). Integrated traffic corridor control using machine learning. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on* (Vol. 4, pp. 3460-3465): IEEE.

Kamal, M. A. S., Mukai, M., Murata, J., & Kawabe, T. (2010a). Ecological driver assistance system using model-based anticipation of vehicle–road–traffic information. *IET Intelligent Transport Systems, 4*(4), 244. doi:10.1049/iet-its.2009.0127

Kamal, M. A. S., Mukai, M., Murata, J., & Kawabe, T. (2010b). On board eco-driving system for varying road-traffic environments using model predictive control. In *Control Applications (CCA), 2010 IEEE International Conference on* (pp. 1636-1641): IEEE.

Kamalanathsharma, R. K. (2012). *Eco-Speed Control Application for In-vehicle Devices using V2I Communication at Rural Intersections*. Paper presented at National Rural ITS Conference Student Paper Competition.

Kamalanathsharma, R. K., & Rakha, H. A. (2013). Multi-stage Dymaic Programming Algorithm for Eco-Speed Control at Traffic signalized Intersections. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on* (pp. 2094-2099): IEEE.

Koenders, E., & Vreeswijk, J. (2008). Cooperative infrastructure. In *Intelligent Vehicles Symposium, 2008 IEEE* (pp. 721-726): IEEE.

Mandava, S., Boriboonsomsin, K., & Barth, M. (2009). Arterial velocity planning based on traffic signal infomation under light traffic conditions. In *12th International IEEE Conference on Intelligent Transportation Systems*.

Mensing, F., Trigui, R., & Bideaux, E. (2011). Vehicle trajectory optimization for application in ECO-driving. In *Vehicle Power and Propulsion Conference (VPPC), 2011 IEEE* (pp. 1-6): IEEE.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. University of Cambridge England.

Wu, C., Zhao, G., & Ou, B. (2011). A fuel economy optimization system with applications in vehicles with human drivers and autonomous vehicles. *Transportation Research Part D: Transport and Environment, 16*(7), 515-524. doi:10.1016/j.trd.2011.06.002

Yang, X., Li, D., & Zheng, P. (2012). Effects of Eco-Driving on Driving Performance. In M. Chu, H. Xu, Z. Jia, Y. Fan & J. Xu (Eds.), *2nd International Conference on Civil Engineering, Architecture and Building Materials* (pp. 2859-2862).

Zhang, T., Kahn, G., Levine, S., & Abbeel, P. (2015). Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search. *arXiv preprint arXiv:1509.06791*.